A project

# ATTENDANCE WITH FACE RECOGNITION

CSE 200: Software Development Project II

**SUBMITTED BY:**


| Name | ID | Intake - Section |
|---|---|---|
| Mahathir Mohammad Rahim | 20245103390 | 53 - 04 |
| Famid Khandoker | 20245103392 | 53 - 04 |
| Ryin Akand | 20245103399 | 53 - 04 |
| Sifath Hossain | 20245103400 | 53 - 04 |
| Nadira Naorin | 20245103401 | 53 - 04 |


**SUPERVISED BY:**


Maharin Afroj

Lecturer

Department of CSE

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BANGLADESH UNVERSITY OF BUSINESS AND TECHNOLOGY**

**(BUBT)**


*15 December,2025*

# Abstract

We are living in a world where everything is automated and linked online. The internet of things, image processing and machine learning are evolving day by day. Many systems have been completely to achieve more accurate results. The attendance system is a typical example of this transition, starting from the traditional signature on a paper sheet to face recognition. This project proposes a method of developing a comprehensive embedded class attendance system using face recognition with showing whether the face of the person is the student for that specified class or not. The system is based on the machine learning algorithm which is to be implemented on python language and using computer or laptop camera for the input image of the students or a normal outer camera can also be used which has to be connected to the system which is programmed to handle the face recognition by implementing the Local Binary Patterns Algorithm LBPS.

# List of Figures

# Table of Contents

# Chapter1

# INTRODUCTION

## 1.1. Problem Specification

Traditional attendance systems such as manual registers, paper sheets, and RFID cards are often time-consuming, unreliable, and prone to manipulation. These methods cannot ensure accurate identification of individuals, as attendance can be marked through proxies or human errors. Moreover, institutions and organizations require real-time monitoring and automated data storage, which traditional methods fail to provide. As a result, maintaining transparency, accuracy, and security becomes challenging.

Therefore, there is a need for an automated, intelligent, and secure system that can identify individuals uniquely and store attendance data accurately without manual effort. A face recognition-based attendance system solves these issues by using biometric technology to detect and verify faces automatically.

## 1.2. Objectives

The main objectives of the Face Recognition Attendance System are:

1. To automate the attendance-taking process using face recognition technology.
2. To eliminate manual errors, proxy attendance, and time delays.
3. To accurately detect and identify a person in real time through a camera.
4. To store attendance records securely in a database for future use.
5. To provide an efficient and user-friendly interface for administrators and users.
6. To generate attendance records, logs, and reports effortlessly.

## 1.3. Flow of the Project

The workflow of the project follows these steps:

**User Registration**
An individual's face images and basic information are captured and stored in the system.

**Model Training**
The system trains a machine-learning model to recognize registered faces.

**Face Detection & Recognition**
During attendance marking, the camera detects the face in real time and compares it with trained data.

**Attendance Marking**
If the face is matched, the system marks attendance automatically with date and time.

**Data Storage**
Attendance records are stored in the database for future retrieval.

**Report Generation**
Admin can view, export, and analyze attendance reports.

## 1.4. Organization Of Project Report

This project report is organized into six chapters:

- **Chapter 1: Introduction**
  Includes the problem statement, objectives, project flow, and overall report structure.
- **Chapter 2: Background**
  Provides details about existing attendance systems and supportive literature related to face recognition technology.
- **Chapter 3: System Analysis & Design**
  Describes tools, technologies, models, diagrams (DFD, Use Case, Context Diagram), database schema, and algorithms.
- **Chapter 4: Implementation**
  Includes interface design, backend development, and system modules explanation.
- **Chapter 5: User Manual**
  Explains hardware/software requirements, user instructions, and panels.
- **Chapter 6: Conclusion**
  Presents the conclusion, limitations, and future enhancements of the system.

# Chapter 2

# BACKGROUND

## 2.1. Existing System Analysis

Before developing the Face Recognition Attendance System, several existing attendance methods were studied. These systems are currently used in schools, universities, offices, and organizations. They range from manual processes to digital biometric systems.

### 1. Manual Attendance System (Paper-Based Register)

This is the traditional method where teachers or supervisors call names and mark presence on a sheet.

**Pros:** Easy to use, no technical knowledge required,no equipment needed except paper and pen.

**Cons:** Time –consuming,especially for large groups. very easy to manipulate(proxy or attendance).Risk of losing registers or misplacing data. No automatic reporting or digital record.

### 2. ID Card / RFID-Based Attendance System

Users swipe their ID cards or RFID tags to record attendance.

**Pros:**Faster than manual systems.Simple to set up and operate.Digital record-keeping is possible.

**Cons:**Cards can be lost, stolen, or swapped between users.Proxy attendance still possible since anyone can carry the card.No biometric verification, so security is limited.

### 3. Fingerprint Biometric System

Attendance is marked using fingerprint scanners.

**Pros:** More secure than ID cards or manual systems. Reduces the chances of proxy attendance. Fast and automatic recording.

**Cons:** Fingerprint scanners may fail due to dirt, sweat, or injuries. Health concerns—multiple users touching the same scanner. Devices can malfunction over time. Slower when many users try to scan at the same time.

### 4. Existing Face Recognition Systems

There are some available digital systems using webcams and face detection algorithms.

**Pros:** Fully contactless and hygienic. High accuracy in identifying users.Fast and convenient for both students and teachers.

**Cons:**Some systems require high-quality cameras.Accuracy may reduce in poor lighting conditions.Costlier compared to manual or RFID systems.Not all systems store data securely or provide full customization.

## 2.2 Supporting Literatures

This project is supported by various theoretical, mathematical concepts. These literatures guided the design and development of the Face Recognition Attendance System.

### 1. Theoretical Concepts

**Biometric Authentication:**
Uses unique human characteristics (face) to identify individuals.

**Pattern Recognition Theory:**
Helps classify and match facial patterns to known data.

**Machine Learning Basics:**
Model training and prediction techniques used for recognizing faces.

### 2. Mathematical Foundations

**Feature Extraction Algorithms:**
Methods like eigenfaces, encodings, and vector space representation are used to convert facial images into numerical data.

**Distance Calculation:**
Euclidean or cosine distance helps match a detected face with stored faces.

**Image Processing Techniques:**
Concepts like pixel matrices, thresholds, and bounding boxes are used.

### 3.Methodological Approaches

**Agile Model:**
Used for planning, designing, coding, testing, and deploying the system.

**Data Flow Modelling:**
Helps visualize how data moves within the system.

**Modular Development:**
System divided into modules like registration, training, recognition, and reporting.

# Chapter 3

# SYSTEM ANALYSIS & DESIGN

## 3.1 Techonology &Tools

To build the Face Recognition Attendance System, a combination of AI libraries, face detection algorithms, and database technologies were used. The selection of tools was based on accuracy, speed, and practical implementation.

### Software Tools & Technologies

**Python:** Main programming language due to strong support for ML and image processing.

**OpenCV:** Used for live video capturing, image preprocessing, and face detection.

**Haarcascade Classifier:** Applied for fast and efficient face detection.

**LBPH Face Recognizer:** Used for face recognition due to high accuracy under varying lighting conditions.

**Mysql Workbench:** Database for storing user details and attendance.

**Tkinter / Web Framework:** For creating GUI.

## 3.2 Model & Diagram

### 3.2.1 Use Case Diagram

The benefit of use case diagrams is mostly based on communication between the request team and the user group. A use case specification document should cover the following areas:

- Actors - participating in and interacting in this use case

- Preconditions - must be met for the use case to work

- Unconditional - defines the various states in which the system is expected to be after it is executed. The Use Case diagram lists the basic events that will occur when the system is executed. It includes all the primary actions that the system must perform.

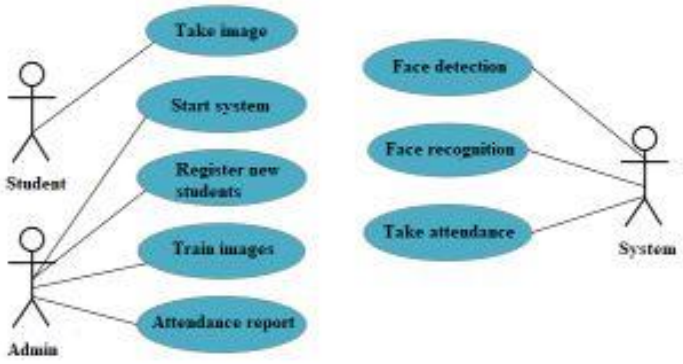


Figure 1:Use Case Diagram

### 3.2.2 Context Level Diagram

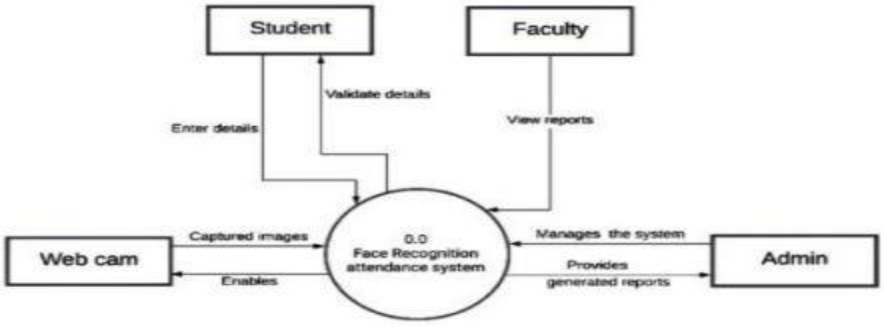


Figure 1: Context Level Diagram

### 3.2.3 Data Flow Diagram

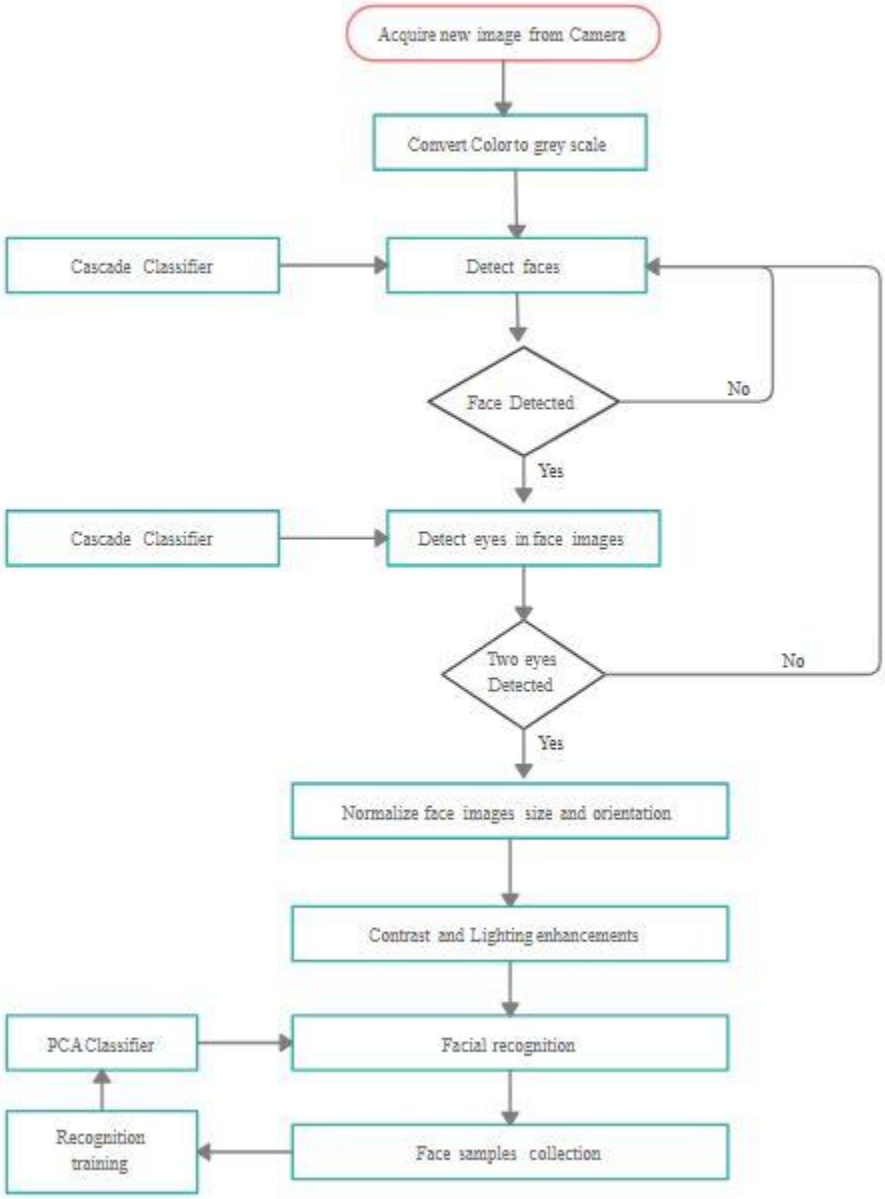


**Figure 3:Data Flow Diagram**

### 3.2.5 Database Schema

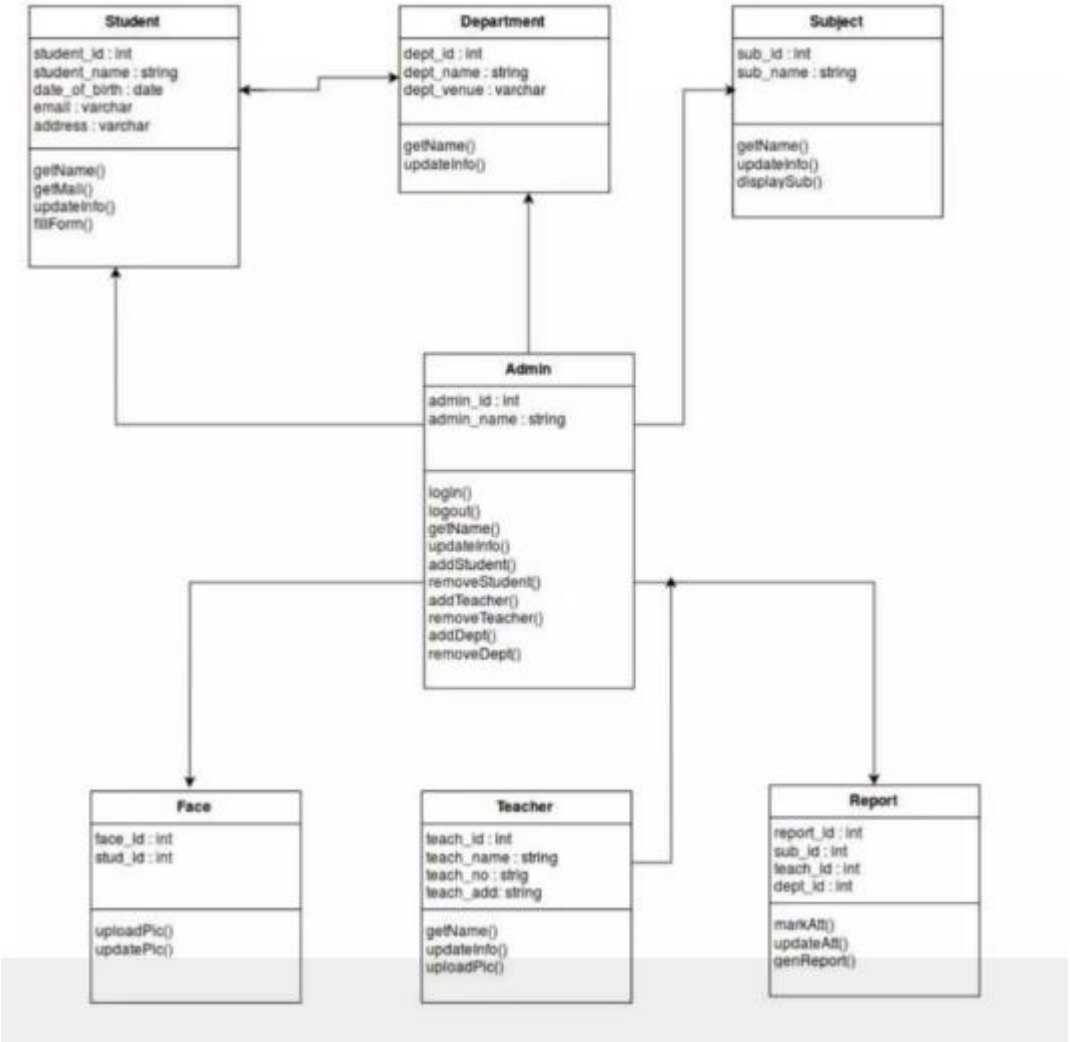


**Figure 4:Database Schema**

### 3.2.6 Flow Chart

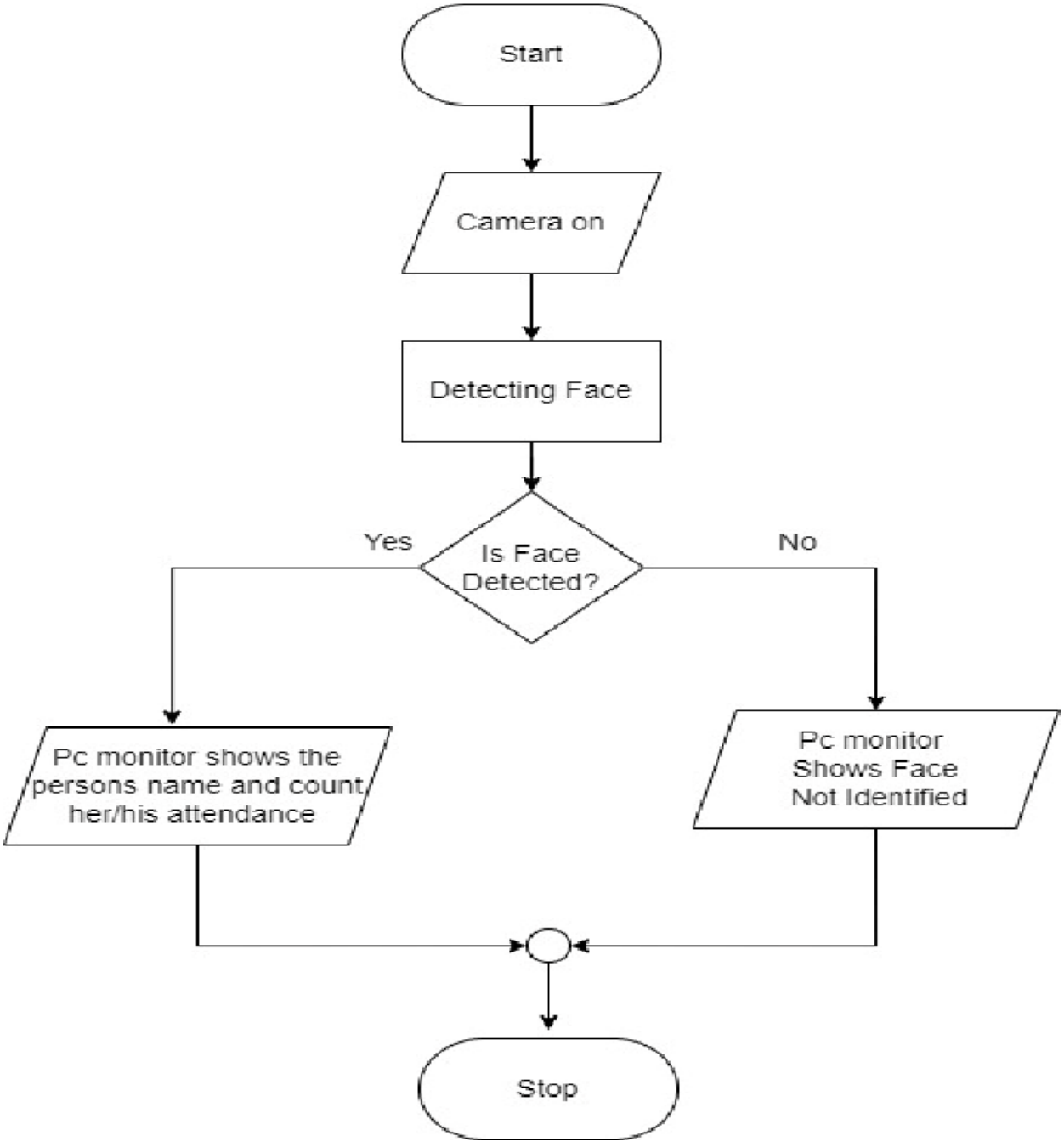


**Figure 5: Flow Chart**

# Chapter 4

# IMPLEMENTATION

## 4.1 INTERFACE DESIGN /FRONT-END

The front-end of the Face Recognition Attendance System is implemented using **Tkinter**, Python's built-in GUI framework. The interface is designed to be simple, responsive, and easy to navigate, ensuring smooth interaction for both administrators and users.

**Home Dashboard:** Provides access to dataset capturing, face training, face recognition, attendance viewing, and system settings.

**Dataset Panel:** Allows users to register their ID and name before capturing facial images.

**Attendance Panel:** Displays real-time attendance logs, including student ID, name, date, and time.

**Training Panel:** Shows training progress using LBPH and updates the model automatically.

**Notification Messages:** All operations (success/failure) show popup messages for better usability.

The UI uses Tkinter widgets such as Labels, Entry Fields, Buttons, Frames, and Canvas for layout, making the system user-friendly and visually clean.

## 4.2 BACK-END

The back-end serves as the core processing unit of the system and is developed using Python, OpenCV**,** Haar Cascade, and LBPH Face Recognizer.

### Major Back-End Operations:

### (a) Haar Cascade for Face Detection

1. Haar Cascade Classifier (`haarcascade_frontalface_default.xml`) is used to detect faces in live camera frames.
2. It identifies facial regions and passes the cropped images to the recognizer.
3. Works efficiently in real-time video streams.

### (b) LBPH (Local Binary Pattern Histogram) for Face Recognition

1. LBPH algorithm extracts texture patterns from facial images.

2. Converts the face into a grid of binary patterns.
3. Produces a histogram that uniquely identifies each person.
4. Robust under lighting changes and low-resolution images.

## (c) Dataset Generation

1. For each registered user, the system captures around **100** facial images**.**
2. Images are stored in `/data` folder with filenames containing user ID.
3. These samples are later used for training the recognition model.

## (d) Model Training

- LBPH recognizer is used (`cv2.face.LBPHFaceRecognizer_create()`).
- Training images and labels are converted into NumPy arrays.
- The trained model is saved as `classifier.xml`.

## (e) Attendance Marking

- When the system identifies a face, it retrieves:
  - Name
  - ID
  - Date
  - Time
  - Attendance Status
- The record is saved automatically into:
  - A **CSV file**, and
  - Optional **MySQL database** (if enabled)

## 4.3 Modules of the System

The system is divided into the following main modules:

**Module 1: Student Registration**

- User enters Name**,** ID**,** and other basic information.
- Data is stored in `student.csv` or MySQL Workbrench table.
- Enables future dataset creation and attendance tracking.

**Module 2: Dataset Creation Module**

- Opens webcam and detects face with Haar Cascade.
- Captures 100 different images of the user.
- Stores them in a folder for training.

**Module 3: Training Module**

- Collects all images from the dataset folder.
- Uses LBPH algorithm to train a recognition model.
- Saves the trained model into `classifier.xml`.
- Shows progress and success popup in UI.

**Module 4: Face Recognition Module**

- Activates the camera.
- Detects the face using Haar Cascade.
- Matches it using the LBPH trained model.
- Marks the attendance in real time.

**Module 5: Attendance Management Module**

- Stores attendance in CSV format `{ID, Name, Date, Time, Status}`.
- Can be viewed in the UI.
- Can export or clear attendance lists if needed.

**Module 6: Database Module**

If MySQL is used:

- Student information table
- Attendance table
- Automatic insertion using Python–MySQL Workbench connector

# Chapter 5

# USER MANUL

## 5.1 System Requirements

### 5.1.1 Hardware Requirement

To run the Face Recognition Attendance System smoothly, the following hardware is recommended:

- **Processor:** Intel Core i3 / Ryzen 3
- **RAM:** 4 GB (8 GB recommended)
- **Storage:** 2 GB free disk space
- **Webcam:** Minimum 720p HD camera
- **Monitor:** 1366×768 or higher resolution
- **Power Backup:** (Optional) UPS for continuous operation

### 5.1.2 Software Requirements

- **Operating System:** Windows 10 / Windows 11 / Linux (Ubuntu)
- **Python Version:** Python 3.7 or later
- **Libraries:**OpenCV,NumPy,Tkinter,Pillow,Pandas,MySQL Workbench Connector (if database enabled)
- **Database:** MySQL Server (optional)
- **IDE / Editor:** VS Code / PyCharm / IDLE

## 5.2 User Interfaces

Several user interfaces are included in this system to ensure a smooth and intuitive user experience.

### 5.2.1 Panel A – Main Dashboard

1. Student Registration
2. Take Image / Dataset
3. Train Face
4. Face Recognition
5. Attendance Sheet
6. Exit

## 5.2.2 Panel B – Dataset & Training Panel

**(a) Dataset Collection Panel**

- User enters Student ID and Name.
- Clicking “Take Image” activates the webcam.
- Haar Cascade detects the face in real-time.
- The system automatically captures 100 images.
- Images are saved in the dataset folder for training.

**(b) Training Panel**

- This module scans all dataset images.
- Trains the model using LBPH Face Recognizer.
- On successful training, a file named `classifier.xml` is generated.
- Progress bar and popup messages indicate training status.
- This is the core module for building the recognition accuracy of the system.

### 5.2.3 Login Credentials Panel

If the system uses admin authentication:

Default Login Credentials:

- Username: admin
- Password: admin123

Login ensures that only authorized users can access sensitive features such as database records and attendance logs.

## 5.2.4 Screenshot


**Figure 6: Main Dashboard**


**Figure 7:Student Details**


**Figure 8: Train Data**

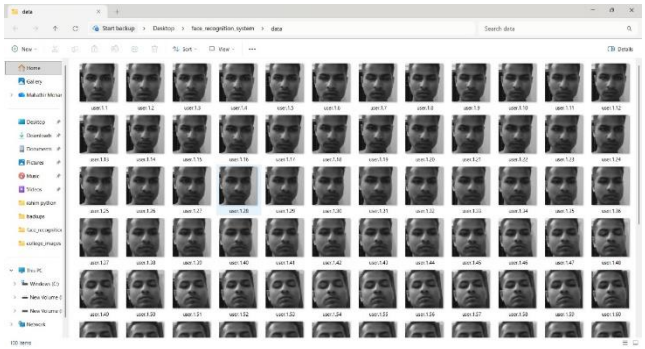

**Figure 9:Image Store**

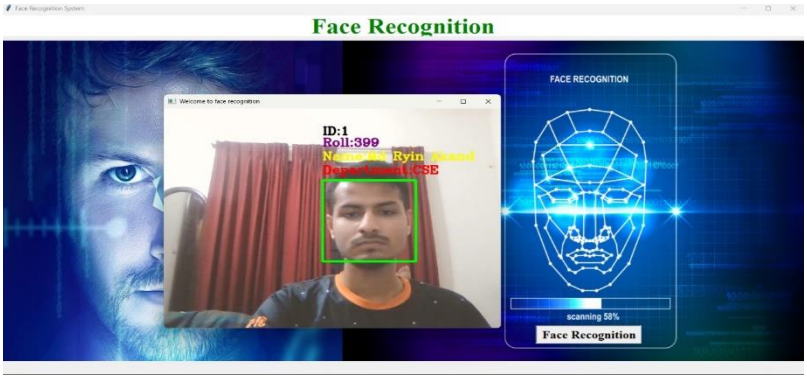


**Figure 10:Face Detector**

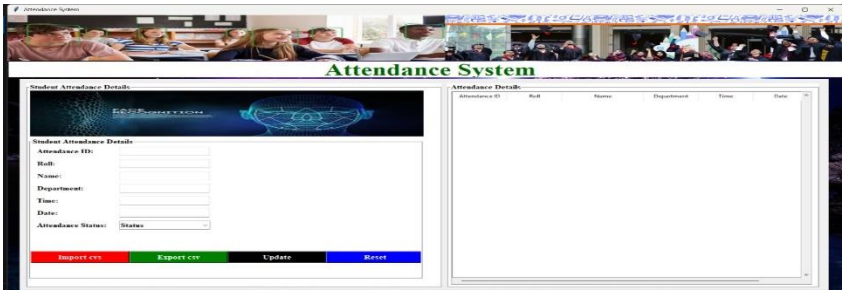


**Figure 11: CSV File Record**

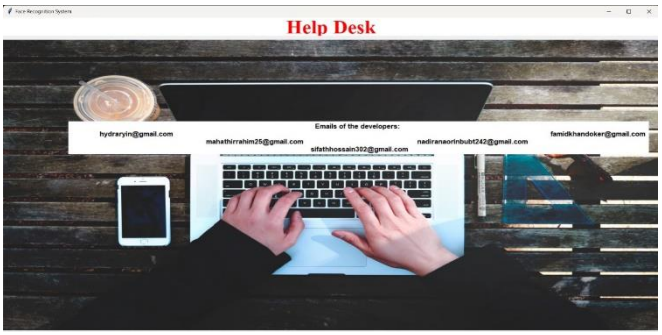


**Figure 12:Help Desk**


**Figure 13: Developers Page**

# Chapter 6

# CONCLUSION

## 6.1 Conclusion

The Face Recognition Attendance System successfully demonstrates the integration of modern computer vision techniques into a real-world attendance management solution. By combining Haar Cascade for face detection and LBPH (Local Binary Pattern Histogram) for face recognition, the system provides an automated, efficient, and user-friendly method for recording attendance. Compared to traditional manual methods such as paper-based registers or ID card scanners, this system ensures greater accuracy, reduces human effort, and minimizes chances of proxy attendance.

The system is implemented using Python, OpenCV, Tkinter, and optional MySQL Workbench database support, making it both flexible and easy to deploy in educational institutions, offices, and organizations. Overall, the project fulfills its objectives and provides a reliable platform for automated attendance management.

## 6.2 Limitations

Despite its successful implementation, the system has several limitations:

**Lighting Sensitivity:**
The accuracy decreases under poor lighting or extreme brightness.

**Camera Quality Dependency:**
Low-resolution or blurry webcams can affect face detection and recognition accuracy.

**Pose Variation Issues:**
If the user looks away, wears a mask, or tilts their head too much, recognition may fail.

**Processing Time:**
Large datasets or low-performance hardware may slow down training and detection.

**Single-Face Recognition at a Time:**
The current model is optimized for recognizing one person per frame rather than crowd detection.

**Basic Security:**
Login and admin security could be improved for high-security environments.

## 6.3 Future Work

To enhance the capability, accuracy, and real-world usefulness of the system, the following improvements can be included in the future:

**Deep Learning-Based Recognition:**
Implementing CNN-based models (e.g., FaceNet, Dlib, or Mediapipe) for higher accuracy.
**Mask-Friendly Recognition:**
Add support for recognizing faces with masks, glasses, or partial occlusions.
**Multi-Face Recognition:**
Enable recognition of multiple individuals at the same time in a single frame.
**Cloud Database Integration:**
Store attendance data in cloud services for remote access and central monitoring.
**Mobile App Integration:**
Develop a mobile version for teachers/admins to check attendance instantly.
**Advanced Anti-Spoofing:**
Add live detection to prevent photo or video-based spoofing attacks.
**Automatic Student Alerts:**
Send SMS/Email alerts for attendance confirmation or absence notifications.
**Dashboard Analytics:**
Add charts and attendance insights for administrators.

## References


- Bradski, G. & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.
- OpenCV Documentation. (2024). *OpenCV Python Tutorials*. Retrieved from https://docs.opencv.org


## Appendix A

This appendix section includes the essential code snippets used in building the Face Recognition Attendance System. It contains the following components:

- A.1.1 Student Registration Module.
- A.1.2 Dataset Capture Module (Using Haar Cascade).
- A.1.3 Training Module (Using LBPH).
- A.1.4 Face Recognition Module.
- A.1.5 Attendance Logging Module.
- A.1.6 Database Connectivity.